Process-parameter-interface model for design management

Chua, D. K. H.; Tyagi, A.; Bok, S. H.; Ling, San

2003

Chua, D. K. H., Tyagi, A., Ling, S., & Bok, S. H. (2003). Process-parameter-interface model for design management. *Journal of construction engineering and management*, 129(6), 653-663.

<https://hdl.handle.net/10356/95764>

[https://doi.org/10.1061/\(ASCE\)0733-9364\(2003\)129:6\(653\)](https://doi.org/10.1061/(ASCE)0733-9364(2003)129:6(653))

© 2003 American Society of Civil Engineers. This is the author created version of a work that has been peer reviewed and accepted for publication by *Journal of Construction Engineering and Management*, American Society of Civil Engineers. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2003\)129:6\(653\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2003)129:6(653))].

Downloaded on 26 Apr 2023 17:57:41 SGT

Process-Parameter-Interface Model for Design Management

D. K. H. Chua, M.ASCE¹; A. Tyagi²; S. Ling³; and S. H. Bok⁴

¹*Associate Professor, Dept. of Civil Engineering, National Univ. of Singapore, Singapore 119260. E-mail: cvedavid@nus.edu.sg*

²*Research Scholar, Dept. of Civil Engineering., National Univ. of Singapore, Singapore 119260. E-mail: engp9818@nus.edu.sg*

³*Associate Professor, Dept. of Mathematics, National Univ. of Singapore, Singapore 119260. E-mail: matlings@nus.edu.sg*

⁴*Senior CAD/CAM Specialist, National Univ. of Singapore, Singapore 119260. E-mail: engboksh@nus.edu.sg*

Abstract

This paper presents a framework for managing the architecture/engineering/construction design process. The framework, called the process-parameter-interface (PPI) model, tries to address the design management issues of improved design process scheduling and efficient collaboration. The model comprises three main components—a design parameter vocabulary, in the form of a *design dictionary*; an *interface*; and an *engine*—that help to address these issues. In addition, the model promotes flexibility of design work flow, enables proactive collaboration, is implementable over the Internet, and is information-centric. The paper concludes with a case study for the design of a conference room to illustrate the functioning of the PPI model and the role of its components.

Introduction

Managing the design process is becoming a core issue in the architecture/engineering/construction (A/E/C) sector due to its strong impact on the entire project. Successful management of the design process is critical to the cost-effectiveness, timeliness, and quality of the entire project. A study suggests that around 20 to 25% of the total construction period is lost due to deficiencies in design (Undurraga 1996), while another study suggests that around 78% of quality problems in A/E/C are attributable to design (Koskela 1992). From the cost viewpoint as well, design-caused defects form the largest category (Josephson and Hammarlund 1996).

Each project in the A/E/C sector is strikingly unique in the way it has to be executed, where it has to be executed, and most importantly, what has to be executed. However, amid all this multifariousness and multiformity, one thing common to all projects is their immense complexities coupled with their fragmentation. A major part of this complexity lies in the design phase of the project, wherein a large number of generally

geographically dispersed project participants (stakeholders) with different objectives and perspectives have to collectively come up with a design solution that satisfies the constraints imposed by the design requirements. Hence design management has assumed significant importance over the years.

At the same time, a new school of thought is diffusing into A/E/C: a growing interest in introducing and applying the principles of lean production (world-class manufacturing) into the various A/E/C processes. The lean production philosophy states that “production consists of conversions and flows, and the overall efficiency of production is attributable to both the efficiency (level of technology, skill, motivation, etc.) of the conversion activities performed, as well as the amount and efficiency of the flow activities through which the conversion activities are bound together” (Koskela 1992, pp. 15–16). Extending the flow view to the design process in A/E/C (Fig. 1) makes it easier to identify the wastes in the design process, such as waiting for information, transformation of information, and inspection. Thus, viewing the A/E/C design process as a flow of information should enable better control over it.

Moreover, the aim of any design process is to produce economically useful information by efficiently generating information of greatest value (Reinertsen 1997). In the context of engineering design, generation of economically useful information can be accomplished by minimizing the rework loops in the design process in terms of their number and length. These rework loops (iterative loops) can best be understood by viewing the design process as a flow of information because the rework is directed at changes in the information.

Accordingly, the model presented in the paper advocates viewing the design process as a flow of information between different design specialists, external authorities, and other stakeholders in the project. The model is named the process parameter interface (PPI) model, and it facilitates transparency, smooth information flow (through efficient design process scheduling), and effective collaboration. These functions are provided by the model components—the *design dictionary*, the *interface*, and the *engine*—which are described in the subsequent sections of the paper.

Review of Research in Design Management

Several studies have been conducted in the field of design management. Some of the more popular models for modeling design are Pugh’s “total design” model, the VereinDeutscherIngenieure (VDI) model of engineering design, and the Pahl and Beitz design model (Austin et al. 1999). The Royal Institute of British Architects (RIBA 1973) plan of work for design team operation is another widely used building design model that models the tasks to be carried out by various design players during each stage but does not model the relationships between the tasks (Austin et al. 1999).

The analytical design planning technique, or AdePT (Austin et al. 2000), has been used to prepare optimized design program (schedules). AdePT models the building design process by representing design activities and their information requirements in a dependency structure matrix. The iteration in the design process is identified and its

constituent activities are scheduled so as to minimize the iterations. Based on the optimized process sequence, a design program is generated. The PPI model presented in this paper addresses the issue of sequencing the design process, as done previously by AdePT, albeit with an explicit emphasis on viewing design as a flow of information (parameter perspective) and not just as a flow of work (process perspective).

Design collaboration frameworks, such as design-agent based collaboration (Khedro et al. 1994) and collaborative construction agents (Jones and Riley 1995), have been developed that are directed toward making collaboration more efficient by means of software agents. These frameworks consist of software agents that interact with one another, thus facilitating design collaboration.

Change management models, namely a layer-based model (Krishnamurthy and Law 1995) and a constraint-checking model (Tiwari and Howard 1994), enable the management of design changes by storing design configurations (design alternatives/states and the constraints acting on them). The management of a highly complex and dynamic process, such as design, requires effective management of changes as and when they are made during the design process. To this end, a proactive approach in managing change is needed in the form of change alerting and resolution.

Frameworks, such as distributed and integrated environment for computer-aided engineering (DICE) (Sriram and Logcher 1993) and the software environment to support early phase in building design (SEED) (Flemming and Woodbury 1995), leverage information technology advancements to manage design. A drawing version control system has also been developed for managing design-drawing versions (Hirasawa and Kataoka 1999).

The key issues in design management identified and researched by academia include optimized design process scheduling, efficient collaboration, and change management. Different frameworks (models) address these basic issues in different ways. Today's highly specialized and dispersed design teams have a need for a framework that, along with addressing these issues, provides flexibility/adaptability, enables proactive collaboration, and is information-centric. The PPI model tries to tie in these features using an Internet platform.

Conventionally, the design work flow is hardwired, meaning that the sequence of design tasks is governed by the work breakdown structure of design. Thus the design processes do not take into account the availability or unavailability of design parameter information needed to execute the design tasks. This leads to situations where the information needed at a certain point of time but not yet produced has to be assumed. In the event that this assumed design parameter information is not congruent with the actual parameter value (generated downstream), a long iterative loop may be created. The PPI model encourages the work flow of design based on the availability of information as much as possible. That is, the design tasks are sequenced so that the need to assume information is minimized, and thus the iterative loops are also minimized, which makes the PPI model flexible and adaptable to varying design processes.

The model enables proactive collaboration for managing design changes by alerting the relevant design specialists as well as managing the iterative loops in the design process by inviting the specialists involved in the loop to enter into collaboration. The model also provides an integrated collaboration-based framework that is implementable on the Internet and thus leverages the speed, power, and distributed environment of the Internet.

Moreover, the model takes an information-centric view of the design process, viewing design as a process of information transformation where abstract information (client requirements) is transformed into specific information (detailed drawings and technical specifications) needed to build the facility as desired. This transformation involves and is facilitated by the flow of information among the design participants (design specialists, clients, and external authorities). In the PPI model, the design work flow is derived from an optimal sequencing of the parametric information required and produced by the design specialists.

The PPI model presented in the paper makes the design process transparent, enables efficient collaboration, and smooths the information flow (through improved design process scheduling). Transparency (in the context of design) can be thought of as a general state in which the design specialists from different disciplines are cognizant of one another's design variables. Efficient collaboration implies the ability of the design participants to share the relevant information at the appropriate times. Smoothing of the information flow is the process of facilitating the flow of information so that the need for assuming information during design is minimized.

Transparency is facilitated by the design dictionary, the interface, and the engine's alert and conflict managers. Efficient collaboration is enabled by the engine's sub managers, namely the alert manager, conflict manager, loop resolution manager, and scheduling manager. Proactive collaboration is achieved by means of dissemination of information to the relevant design participants. Smooth information flow is enabled by the engine's scheduling manager, and lastly, the management of design change is facilitated by the engine's alert and conflict managers.

The following section gives a brief description of the conventional A/E/C design process, and then the PPI model is introduced and developed in detail.

Conventional Design Process in A/E/C

The process of design is a means of achieving the most suitable solution to a particular design problem, taking into account multiple constraints arising due to the requirement for satisfactory design from multiple perspectives. The design process is generally iterative in nature, involving the following broad phases: recognition of needs, development of requirements, conceptual design, and detailed design. Design in the A/E/C sector is generally architect driven, for the architect is responsible for translating abstract client requirements into a preliminary project layout that then forms the foundation for preparation of detailed designs by the different design specialists. Conventional design processes were studied at a design/build organization to gain better insight into the processes and to implement the proposed design management model.

The study revealed two levels of information exchange occurring in the A/E/C design process: broad level and detailed design level. At the broad level of information exchange, two-way interaction is observed between the architect and each of the design specialists for general understanding of the client's intent and the design intent as interpreted by the architect. System-level tuning is performed at this stage, and the broad-level design parameters are established that will govern the detailed design. As an example of the parameters that govern the detailed design, consider the design specialists deciding upon a key parameter such as the type of roof. The architect may suggest use of a particular type of roof, but the contract engineer may point out that the delivery time for that particular roof is long and that the roof is expensive and without warranty. The architect would then resuggest another roof design, which would alter the loads to be considered by the structural designer. Broad issues such as these are discussed at this level.

The detailed design level follows the broad level of information exchange. Each of the design specialists proceeds with his or her design, based on the broad design guidelines formulated at the broad level (broad-level parameters). Interaction (information exchange) among the designers at this level is limited and mainly passive, triggered when a design specialist faces a potential conflict situation with another design specialist. For example, an electrical design specialist, while performing detailed design of the lightning protection system, may need to know about the rebar layout in a column before designing the layout for cast-in pipes and thus would interact with the structural designer to avoid the conflict. Information exchange needed to avoid a design conflict is generally inevitable. Another reason for interaction among the design specialists at this level is the need to confirm their designs with each other, which is usually done after the drawings have been prepared by them. Thus, incorporating any changes would imply expensive rework, but at times this form of interaction is unavoidable.

Fig. 2 depicts the design process described above. Interaction has been shown only among two of the design processes, namely structural and HVAC, for the sake of clarity. Subprocesses within the structural and HVAC design disciplines are all unaware of each other's design parameters, particularly in the detailed design phase, thus leading to an overall design process that is highly susceptible to conflicts and consequently to numerous rework loops. Moreover, the conflict among the structural and HVAC designs is discovered after the drawings/specifications have been prepared, leading to lengthy rework loops 1 and 2.

Process-Parameter-Interface Model

The long rework loops in the preceding section may be resolved through improved transparency, sequencing, and collaboration in the design process. The PPI model (Fig. 3) tries to promote these features by managing the publishing and flow of key design parameters. These parameters are the key variables needed for analysis and design performed by the specialists; they are the driving factors of the design process and should be exchanged as early as possible, instead of after the design drawings have been prepared. The main components of the model—the *interface*, the *engine*, and the *design dictionary*—are described next.

Interface

With increasing specialization in the A/E/C design, multidisciplinary teams, often from different organizations, are becoming increasingly common. Most of these design specialists have their own proprietary approach to design, and although they interact with each other to resolve conflicts or to improve the design, they are not always able to share their design methodology confidentially with other specialists.

In the PPI model, each design specialist needs to disclose only the key driving parameters he or she requires to perform the design functions and parameters that he or she generates, corresponding to the design tasks for which he or she is responsible. This is achieved by means of the *interface* (Fig. 3), which acts as the medium through which the design specialists can share parameter information (in the form of an information requirement table and information production table). The interface thus makes the design process transparent with regard to the information dependencies among the design participants, without compromising the confidentiality of the design processes.

As an illustration of the role of the interface, consider the design of a conference room in which the structural, HVAC, and lighting specialists have to collaborate to derive a consistent design. The design processes of the specialists are described below.

1. Structural Design Process
 - a. Determination of free height, based on architectural requirements, depth of beam, and space requirements for lighting and HVAC ducting.
2. HVAC Design Process
 - a. Heating loads determination
 - (1) Determine air quality requirements based on client specifications or otherwise acceptable levels; and
 - (2) Determine heating loads and airflows using HVAC manuals.
 - b. Layout
 - (1) Lay out duct system on floor plan, accounting for the direction of joists, roof hips, firewalls, and other potential obstructions. Determine register locations and types, duct lengths, and connections required to produce layout given construction constraints;
 - (2) Size duct system according to HVAC manual calculation procedures; and
 - (3) Size HVAC equipment to sensible load using HVAC manual procedures.
3. Lighting design process
 - a. Determine lighting requirements based on client specifications or otherwise acceptable levels;
 - b. Prepare lighting layout for uniform illumination requirements or any other requirements as planned by architect or the client; and
 - c. Select the type of lighting based on available space, levels of illumination desired, and cost considerations.

The three design specialists do not need to share their design processes and methodologies, but only to publish the information they require and produce in the form of an information requirement table and an information production table, respectively, through the *interface*, as shown in Fig. 4. The corresponding user interface of the

prototype has been depicted in Fig. 5, showing the interface screens for the three design specialists. This information is then processed by the PPI model engine, details of which are provided in the following section.

PPI Model Engine

The *engine*, which facilitates the functional logic for the PPI model, consists of a collaboration manager that has four submanagers: the scheduling manager, loop resolution manager, conflict manager, and alert manager, as shown in Fig. 6. These provide design management functions such as design workflow sequencing, design loop collaboration, conflict resolution, and parameter value alerting, respectively, as illustrated in the figure. The latter three functions make the collaboration efficient. The functionality provided by the engine (through the submanagers) is based on the information dependencies mapped out by the scheduling manager in the form of an information-based design dependency matrix.

The information requirement and production tables are synthesized into a matrix form, comprising all the key parameters, by the scheduling manager, who then triangularizes this matrix to yield an information-based design dependency matrix. (Triangularization is the process of reordering the elements of the matrix so that the number and size of the loops is minimized.) This matrix governs the design workflow and the other design management functions facilitated by the *engine*. Fig. 7 shows the information-based design dependency matrix generated earlier by the PPI model engine for the design case illustrated in Fig. 4. The key design parameters involved in the three processes— structural, HVAC, and lighting—are written on the left and top of the matrix. The asterisks in the cells of the matrix denote the dependency of the corresponding row parameter on the column parameter; that is, the generation of the row parameter depends on the column parameter(s).

As mentioned above, the scheduling manager uses the parameters published through the interface to generate the design parameter sequence in the form of an information-based design dependency matrix. The loop resolution manager determines the rework (iterative) loops from the information-based design dependency matrix and invites the design specialists producing the parameters involved in the loop to enter into a collaboration so as to avoid potential rework cycles. The conflict manager keeps track of any request for change (RFC) issued by any design specialist for a particular parameter. If any new RFC is found, the conflict manager notifies and invites the concerned design specialist(s) (those that require the parameter for their designs) to enter into a collaboration to resolve the conflict. The alert manager checks for newly published parameter values and uses the information-based design dependency matrix to determine the parameters that depend upon these newly published parameters. It then alerts the owner(s) of the parameters about the change. The alert manager can then be attached with a time-based schedule (parameter generation schedule) and inform the parameter owners of the latest time by which they must produce their parameters and subsequently pull the parameter values.

The submanagers determine the design parameter owners from the design dictionary, which is elaborated in the following section.

Design Dictionary

The design dictionary acts as a repository for key design parameters that govern the execution of design. It imparts transparency by enabling the design specialists to view each other's design parameter requirements, and it is also used to check the validity of the information production table data published by a design specialist through the interface. In case a parameter required by a design specialist that is published through the interface has not been defined in the design dictionary, the interface manager shall prompt the design specialists to enable one of them to accept ownership of the parameter. Thus the *design dictionary* ensures that all parameters have been owned. On the other hand, if the parameter to be produced by a design specialist and published through the interface has not been defined in the design dictionary, the design specialist shall be prompted to enter the parameter details.

Each of the design parameters is also characterized by some attributes that yield a better insight into the design process. These attributes have been classified as static and dynamic. The static attributes are fixed, while the values of dynamic attributes may mutate during the design process. These attributes are discussed next.

The *static attributes* include *estimability*, *volatility*, and *ownership*.

1. *Estimability* describes the degree of ease with which the value of a parameter can be assumed or estimated in case the parameter value is unavailable. If the parameter has a well-defined and narrow range of values, it is supposed to have high *estimability*. *Estimability* has been graded as low, medium, and high. This attribute can give a fair idea as to whether a particular design task can be started even before all the parameters required by it are available. For instance, the value of a parameter with high *estimability* may be assumed even before its confirmation.
2. *Volatility* is the degree of potential for change of the design parameter and is generally due to an external environment, such as building authorities or other agencies. If the parameter has a high chance of being changed, say by an external authority, it is supposed to be highly volatile. Like *estimability*, *volatility* has also been graded as low, medium, and high. *Estimability* and *volatility* together form an important indicator for the downstream participants in determining the potential likelihood of a design change upstream.
3. *Ownership* refers to the design specialist or owner who produces a particular parameter. The loop resolution manager, conflict manager, and alert manager of the *engine* determine the *owners* of the relevant parameters from the design dictionary before inviting, notifying, or alerting them, respectively.

The *dynamic attributes* include *state* and *iteration count*.

1. *State* denotes the ability of a particular design parameter to undergo a change in its value at any particular instant of time. A parameter may exist either in a liquid or frozen state. A design parameter in a frozen state is immutable, that is, it cannot be altered any further but can still be changed by an external agency. On the other hand,

a parameter in a liquid state is open to any kind of negotiation among concerned design specialists, that is, it is mutable. A frozen state implies that the downstream design can be undertaken with greater confidence.

2. The *iteration count* for a parameter, at any instant in time, denotes the number of times the parameter value has been iterated upon until then. This attribute, captured together with the reason for iteration, can act as a knowledge base to manage the rework in similar scenarios in the future. However, the learning aspect is beyond the scope of this paper.

Case Study: Design of a Conference Room

A case study is presented here to illustrate the working of the PPI model. It is based on interactions with design specialists from two large companies, one of which is a multinational organization. The case concerns the design of a conference room (rectangular in shape with two external and two partition walls) in an existing multistory building. The study involves eight design participants: the client, external authorities, the architect, and the structural, HVAC, mechanical, electrical, and plumbing design specialists, spanning six design disciplines and comprising 37 key design parameters.

The key design parameters associated with the design are identified and have been illustrated in Fig. 8. These parameters, along with their attributes, are defined in the *design dictionary*. The design specialists publish these parameters in the form of information requirement and production tables, respectively, through the PPI model *interface*.

The information requirement and production tables are synthesized into a matrix form (Fig. 9) by the scheduling manager. This matrix shows the interdependencies among the key design parameters. The parameter sequence in the matrix is based on the typical sequence in which the design participants are involved in the design process. In the matrix in Fig. 9, the design parameters—*volume of room (V)*, *height of duct (HD)*, *shape of duct (SD)*, *alignment of HVAC ducts (AD)*, *overhead sprinkler pipe diameter (SPD)*, and *C/C nozzle spacing (NS)*—are dependent upon the parameters produced downstream (relative to them). This introduces rework into the design process, where some or most of the aforementioned design parameters are produced by assuming the value of the downstream parameters (yet to be produced) that they require.

If the assumed values for the downstream parameters do not match the actual values of the downstream parameters, some or all design parameters need to be revised. For example, the parameter AD depends upon two parameters—*artificial lighting pattern design (ALPD)* and *location of electrical nodes and plugs (LNP)*—that are produced further downstream in the design process. Hence the HVAC design specialist (responsible for generating AD) assumes ALPD and LNP, which are produced later (downstream) by the architect and the electrical design specialist, respectively. If either ALPD or LNP or both differ from their assumed value(s), rework shall have to be done by the HVAC design specialist, the architect, or the electrical design specialist, or by all three design specialists.

The PPI model engine (scheduling manager) resequences the parametric flow of Fig. 9 through triangularization to yield an optimized sequence in the form of an information-based design dependency matrix (Fig. 10). This matrix comprises only one loop—the interdependency loop (*HD-FH-V-IF-ALID*)—and all other loops have been eliminated. This loop remains even after triangularization has been done and hence is an inevitable loop that can and should be resolved through collaboration between the architect and the HVAC design specialist.

This resulting matrix (Fig. 10) is used by the submanagers of the collaboration manager to perform their respective functions. The loop resolution manager, for instance, will look for all the loops in the information-based design dependency matrix and facilitate the collaboration of involved parties before the design progresses downstream. In this case, the loop resolution manager determines the loop *HD-FH-V-IF-ALID* and then determines the design specialists producing these parameters through the *design dictionary*, namely the architect (who produces *FH*, *V*, *IF*, and *ALID*) and the HVAC design specialist (who produces *HD*), and relays a message to each of them, inviting them to enter into collaboration to resolve the loop.

The conflict manager uses the information-based design dependency matrix to determine all the parameters downstream of a design parameter for which an RFC has been issued. For example, say, the conflict manager determines (from the RFC data) that an RFC has been issued for the design parameter *AFD* (audio facilities design). It will then determine the parameter(s) downstream of *AFD* from the information-based design dependency matrix that will be affected. The electrical design specialist, for example, is the owner of the immediate successors, *LNP* and *WS*.

The alert manager also uses the information-based design dependency matrix to determine the design parameter(s) succeeding the design parameter, whose value has been published for the first time. For example, in the event of the value of the parameter *AFD* being published by a design specialist for the first time, the alert manager will determine this parameter (and its value) from the parameter value data and look up the information-based design dependency matrix for its succeeding parameter(s), *LNP* and *WS*. The owner(s) of these parameters (electrical design specialist) is or are then determined from the design dictionary, and the newly published value of *AFD* is relayed to them.

The case study has illustrated the working of the PPI model and the role of the engine in improving design process scheduling and making the collaboration efficient. The following concluding section summarizes the paper and discusses the future scope of work.

Conclusions

A framework for management of design, in the form of a PPI model implemented over the Internet, has been provided in this paper. The model makes the design process lean by reducing the design time through improved scheduling (reducing reworks) and efficient collaboration. It also provides flexibility/adaptability, enables proactive collaboration,

and is information-centric.

The model encourages and aims to exploit the idea of parameter-based design management, wherein the flow of design parameter information is central to the management of design processes. Adoption of the information-centric view in A/E/C is still in its primitive stages, though the aviation industry (Boeing) has started using it to manage its new product development operations.

This paper closes with a detailed case study for the design of a conference room to illustrate the working of the PPI model and explain the role of its components. A prototype for the model has been developed using Java/XML to show its implementation.

It is hoped that the PPI model makes a meaningful contribution to the management of design in A/E/C. The model should be applicable to all projects where it is feasible to obtain the information requirements from the project participants. The information definition, however, may vary depending on the design stage. This poses the challenge of creating the design dictionary. A study of the organization and structuring of the parameter data in the design dictionary to form intelligent templates can ease the process of creating it.

References

- Austin, S., Baldwin, A., Li, B., and Waskett, P. (1999). "Analytical design planning technique: A model of the detailed building design process." *Des. Stud.*, 20(3), 279–296.
- Austin, S., Baldwin, A., Li, B., and Waskett, P. (2000). "Analytical design planning technique (AdePT): A dependency structure matrix tool to schedule the building design process." *Constr. Manage. Econom.*, 18(2), 173–182.
- Flemming, U., and Woodbury, R. (1995). "Software environment to support early phase in building design (SEED): Overview." *J. Archit. Eng.*, 1(4), 147–152.
- Hirasawa, G., and Kataoka, M. (1999). "Development of a drawing version control system, used to support alternative designs and ensure the integrity of external references." *Proc., 2nd Int. Conf. on Concurrent Engrg. in Construction*, Int. Council for Research and Innovation in Building and Construction, Rotterdam, The Netherlands.
- Houvila, P., Koskela, L., and Lautanala, M. (1997). "Fast or concurrent: The art of getting construction improved." *Lean Construction*, L. F. Alarcon, ed., Balkema, Rotterdam, The Netherlands, 143–159.
- Jones, B., and Riley, M. (1995). "Collaborative construction agents." *Computing in civil engineering*, J. P. Mohsen, ed., ASCE, New York, 1316–1323.
- Josephson, P., and Hammarlund, Y. (1996). "Kvalitetsfelkostnader pa 90-talet-en studie av sju byggprojekt." *Del I. Rep. 49*, Chalmers Tekniska Hogskola, Institutionen for byggnadsekonomi och byggnadsorganisation, Göteborg, Sweden (in Swedish).
- Khedro, T., Genesereth, M. R., and Teicholz, P. M. (1994). "A framework for collaborative distributed facility engineering." *Computing in civil engineering*, K. Khozeimeh, ed., ASCE, New York, 1489–1496.
- Koskela, L. (1992). "Application of the new production philosophy to construction." *Technical Rep. No. 72*, Center for Integrated Facility Engineering, Stanford Univ., Stanford, Calif.
- Krishnamurthy, K., and Law, K. H. (1995). "Change management for collaborative engineering." *Computing in civil engineering*, J. P. Mohsen, ed., ASCE, New York, 1110–1117.
- Reinertsen, D. G. (1997). *Managing the design factory*, Free Press, New York.
- RIBA Plan of Work for Design Team Operation*. (1973). Royal Institute of British Architects, RIBA Publications, London.
- Sriram, D., and Logcher, R. (1993). "The MIT Dice project." *Computer*, 64–65.

Tiwari, S., and Howard, H. C. (1994). "Constraint management on distribute AEC configurations." *Computing in civil engineering*, K. Khozeimeh, ed., ASCE, New York, 1607–1614.

Undurraga, M. (1996). "Construction productivity and housing financing." *Seminar and Workshop*, Interamerican Housing Union, Ciudad de Mexico, D.F., Mexico.

List of Figures

- Fig. 1 Design as flow of information (Houvila et al. 1997)
- Fig. 2 Conventional design process in A/E/C
- Fig. 3 Process-parameter-interface model
- Fig. 4 Information requirement and production tables for structural, HVAC, and lighting processes
- Fig. 5 Data capturing through PPI model interface
- Fig. 6 PPI model engine
- Fig. 7 Information-based design dependency matrix
- Fig. 8 Key design parameters
- Fig. 9 Matrix representation of typical design work flow
- Fig. 10 Information-based design dependency matrix

Fig.1

Fig.2

Fig.3

Structural Process

Information Requirement Table

Tasks	Parameters	
T _{s1}	Beam Depth (H _b)	Duct Size (D)

Information Production Table

Tasks	Parameters
T _{s1}	Free height (H _f)

HVAC Process

Information Requirement Table

Tasks	Parameters	
T _{h1}	Heating load (Q)	Free height (H _f)
T _{h2}	Length (L)	Room width (B)

Information Production Table

Tasks	Parameters
T _{h1}	Duct Size (D)
T _{h2}	Duct Spacing (S)

Lighting Process

Information Requirement Table

Tasks	Parameters		
T _{l1}	Illumination Req'd. (I)	-	-
T _{l2}	Length (L)	Room width (B)	Duct Size (D)

Information Production Table

Tasks	Parameters
T _{l1}	Lighting Equip. Size (H _l)
T _{l2}	Lighting Spacing (S _l)

Fig.4

Fig.5

Step	Description
1	Information Requirement/Production tables fed into the model through the <i>Interface</i>
2	Scheduling manager synthesizes these tables into Information-based Design Dependency Matrix
LR 3	Loop Resolution manager determines the loops from the matrix produced in Step 2
A 3	Alert manager determines first-time published parameter value from Parameter Value Data
C 3	Conflict manager determines RFC (s) from RFC data
4	Loop Resolution, Alert and Conflict Managers determine the parameter owners from the <i>Design Dictionary</i>
5	Loop Resolution, Alert and Conflict Managers prompt the parameter owners determined in Step 4

Fig.6

Fig.7

Symbol	Parameter Name (Owner)	Symbol	Parameter Name (Owner)
L	Building Loads (C)	EWM	External Wall Material (A)
S	Number of storeys (C)	PWM	Partition Wall Material (A)
N	Number of people to cater for (C)	AFD	Audio Facilities Design (A)
FPR	Furniture Placement Requirements (C)	VFD	Video Facilities Design (A)
OCR	Orientation of Conference Room (C)	PD	Pantry Design (A)
ACR	Acoustic Requirements (C)	LNP	Location of Electrical Nodes & Plugs (E)
ST	Slab Thickness (S)	WS	Wiring Space Requirement (E)
BD	Beam Depth (S)	HD	Height of Duct (HVAC)
CW	Column Width (S)	SD	Shape of Duct (HVAC)
EWT	External Wall Thickness (S)	AD	Alignment of HVAC Ducts (HVAC)
AVR	Air Ventilation Requirements (A)	SPD	Overhead sprinkler pipe diameter (M)
V	Volume of Room (A)	SS	Overhead sprinkler pipe spacing (M)
LR	Lighting Requirements (A)	NS	c/c Nozzle Spacing (M)
PWT	Partition Wall Thickness (A)	LD	Location of Fire protection devices (M)
DWA	Door & Window Arrangement (A)	WTL	Water Tap Location (P)
IF	Interior Finishing (A)	WPA	Water Pipe Alignment (P)
ALID	Artificial Lighting Intensity Design (A)	WPD	Water Pipe Diameter (P)
ALPD	Artificial Lighting Pattern Design (A)	FCR	Fire Code Requirements (EA)
FH	Free Height (A)		

C:	Client	HVAC:	HVAC design specialist
S:	Structural design specialist	M:	Mechanical design specialist
A:	Architect	P:	Plumbing design specialist
E:	Electrical design specialist	EA:	External Authority

Fig.8

C: Client
 S: Structural design specialist
 A: Architect
 E: Electrical design specialist
 HVAC: HVAC design specialist
 M: Mechanical design specialist
 P: Plumbing design specialist
 EA: External Authority

Fig.9

		L	S	N	FPR	OCR	ACR	FCR	AVR	LR	PWT	EWM	DWA	ST	BD	CW	EWT	PWM	VFD	PD	HD	FH	V	IF	ALID	WTL	SD	ALPD	AFD	LD	SPD	SS	NS	LNP	WS	AD	WPD	WPA	
C	L																																						
C	S																																						
C	N																																						
C	FPR																																						
C	OCR																																						
C	ACR																																						
E	A	F	C	R																																			
A	A	V	R																																				
A	L	R																																					
A	P	W	T																																				
A	E	W	M																																				
A	D	W	A			*			*	*																													
S	S	T	*																																				
S	B	D	*																																				
S	C	W	*																																				
S	E	W	T	*																																			
A	P	W	M			*																																	
A	V	F	D			*																																	
A	P	D			*																																		
H	H	D		*	*					*	*	*		*		*	*	*			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
A	F	H						*													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
A	V							*												*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
A	I	F			*	*			*				*		*		*			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
A	A	L	I	D					*											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
P	W	T	L																	*							*												
H	S	D		*	*				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
A	A	L	P	D		*			*											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
A	A	F	D		*	*	*	*	*				*		*		*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
M	L	D				*									*					*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
M	S	P	D			*			*				*		*		*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
M	S	S				*			*				*		*		*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
M	N	S				*			*				*		*		*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
E	L	N	P																*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
E	W	S													*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
H	A	D													*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
P	W	P	D		*																						*											*	
P	W	P	A																																	*		*	

Loop (HD-FH-V-IF-ALID)

Fig.10